CLAIMS:

We claim:

1. A checkpoint processor configured for coupling to individual Web services through a Web services engine, said checkpoint processor comprising:

checkpoint logic programmed to store checkpoint data for the individual Web service instance invocations;

restart logic programmed to restore said stored checkpoint data to a replacement for failed ones of the individual Web service instance invocations; and,

cleanup logic programmed to removed said stored checkpoint data for concluded, non-failed ones of the individual Web service instance invocations.

2. The checkpoint processor of claim 1, further comprising logic for identifying an asynchronous correlator for each one of the individual Web service instance invocations and for storing said asynchronous correlator in association with corresponding ones of said stored checkpoint data.

3. A method for managing checkpoints in a Web application, the method comprising the steps of:

storing a state object for an invocation of a requesting Web service instance; and,

responsive to a failure in said Web service instance, restarting a replacement Web service instance and providing said state object to a replacement Web service instance for said requesting Web service instance.

4. The method of claim 3, wherein said storing step further comprises storing a unique identifier for said requesting Web service instance along with said stored state object.

5. The method of claim 4, wherein said storing step further comprises the steps of:

identifying an asynchronous correlator for said invocation; and,

storing said identified asynchronous correlator along with said stored state object.

6. The method of claim 3, wherein said storing step comprises the steps of:

detecting a notable event in said Web service instance; and,

responsive to said detection, storing a state object for an invocation of a requesting Web service instance.

7. The method of claim 3, wherein said storing step comprises the step of periodically storing a state object for an invocation of a requesting Web service instance.

8. The method of claim 4, wherein said step of providing further comprises the step of providing said unique identifier to said replacement Web service instance.

9. The method of claim 5, wherein said step of providing further comprises the step of providing said asynchronous correlator to said replacement Web service instance.

10. The method of claim 3, further comprising the step of discarding said stored state object when said Web service invocation has completed said invocation nominally.

11. The method of claim 3, further comprising the step of storing state data for a handler chain managing said Web service instance.

12. The method of claim 3, further comprising the steps of:

storing a residency indicator for said Web service instance invocation; and,

registering at least one selected event which when received causes an initiation of said restarting and providing steps.

13. The method of claim 3, wherein said step of restarting comprises the steps of:

determining whether an existing Web service instance can act as said replacement Web service instance; and,

if an existing Web service instance cannot be located which can act as said replacement Web service instance, instantiating a replacement Web service instance.

14. A machine readable storage having stored thereon a computer program for managing checkpoints in a Web application, the computer program comprising a routing set of instructions for causing the machine to perform the steps of:

storing a state object for an invocation of a requesting Web service instance; and, responsive to a failure in said Web service instance, restarting a replacement Web service instance and providing said state object to a replacement Web service instance for said requesting Web service instance.

15. The machine readable storage of claim 14, wherein said storing step further comprises storing a unique identifier for said requesting Web service instance along with said stored state object.

16. The machine readable storage of claim 15, wherein said storing step further comprises the steps of:

identifying an asynchronous correlator for said invocation; and,

storing said identified asynchronous correlator along with said stored state object.

17. The machine readable storage of claim 14, wherein said storing step comprises the steps of:

detecting a notable event in said Web service instance; and,

responsive to said detection, storing a state object for an invocation of a requesting Web service instance.

18. The machine readable storage of claim 14, wherein said storing step comprises the step of periodically storing a state object for an invocation of a requesting Web service instance.

19. The machine readable storage of claim 15, wherein said step of providing further comprises the step of providing said unique identifier to said replacement Web service instance.

20. The machine readable storage of claim 16, wherein said step of providing further comprises the step of providing said asynchronous correlator to said replacement Web service instance.

21. The machine readable storage of claim 14, further comprising the step of discarding said stored state object when said Web service invocation has completed said invocation nominally.

22. The machine readable storage of claim 14, further comprising the step of storing state data for a handler chain managing said Web service instance.

23. The machine readable storage of claim 14, further comprising the steps of:

storing a residency indicator for said Web service instance invocation; and,

registering at least one selected event which when received causes an initiation of said restarting and providing steps.

24. The machine readable storage of claim 14, wherein said step of restarting comprises the steps of:

determining whether an existing Web service instance can act as said replacement Web service instance; and,

if an existing Web service instance cannot be located which can act as said replacement Web service instance, instantiating a replacement Web service instance.